

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ГРОЗНЕНСКИЙ  
ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ  
АКАДЕМИКА М.Д. МИЛЛИОНЩИКОВА»**

---

Кафедра «Информационные системы в экономике»

**Т.Р. Магомаев**

**Методические указания к практическим работам  
по дисциплине «Моделирование бизнес-процессов»**

(направление подготовки 09.03.03 – Прикладная информатика,  
направленность (профиль) – Прикладная информатика в экономике,  
квалификация - бакалавр)

**Грозный 2020**

## Содержание

<b>1.Методология функционального моделирования SADT .....</b>	<b>3</b>
<b>2.Нотация IDEF0 .....</b>	<b>5</b>
Задания к выполнению .....	9
Контрольные вопросы.....	9
<b>3.Методология IDEF3 .....</b>	<b>11</b>
Задания к выполнению .....	17
Контрольные вопросы.....	18
<b>4.Методология ARIS .....</b>	<b>19</b>
<b>5.Описание нотации ARIS eEPC .....</b>	<b>27</b>
Контрольные вопросы.....	30
Задания к выполнению .....	30
<b>6.Нотация DFD.....</b>	<b>32</b>
Задания к выполнению .....	37
Контрольные вопросы.....	38

## 1. Методология функционального моделирования SADT

SADT – одна из самых известных и широко используемых систем проектирования. SADT – аббревиатура слов Structured Analysis and Design Technique (Технология структурного анализа и проектирования) – это графические обозначения и подход к описанию систем. Дуглас Т. Росс ввел их почти 20 лет назад (SADT была создана и впервые опробована на практике в период с 1969 по 1973 г.).

SADT – это методология, разработанная специально для того, чтобы облегчить описание и понимание искусственных систем, попадающих в разряд средней сложности. Военно-воздушные силы США использовали методику SADT в качестве части своей программы интеграции компьютерных и промышленных технологий (Integrated Computer Aided Manufacturing, ICAM) и назвали ее IDEF0 (Icam DEFinition). Целью программы ICAM было увеличение эффективности компьютерных технологий в сфере проектирования новых средств вооружений и ведения боевых действий. Одним из результатов этих исследований являлся вывод о том, что описательные языки не эффективны для документирования и моделирования процессов функционирования сложных систем. Подобные описания на естественном языке не обеспечивают требуемого уровня непротиворечивости и полноты, имеющих доминирующее значение при решении задач моделирования.

В рамках программы ICAM было разработано несколько графических языков моделирования, которые получили следующие названия:

- Нотация IDEF0 - для документирования процессов производства и отображения информации об использовании ресурсов на каждом из этапов проектирования систем.
- Нотация IDEF1 - для документирования информации о производственном окружении систем.
- Нотация IDEF2 - для документирования поведения системы во времени.
- Нотация IDEF3 - специально для моделирования бизнес-процессов.

Нотация IDEF2 никогда не была полностью реализована. Нотация IDEF1 в 1985 году была расширена и переименована в IDEF1X. Методология IDEFSADT, нашла применение в правительственных и коммерческих организациях, поскольку на тот период времени вполне удовлетворяла различным требованиям, предъявляемым к моделированию широкого класса систем.

В начале 1990 года специально образованная группа пользователей IDEF (IDEF Users Group), в сотрудничестве с Национальным институтом по стандартизации и технологии США (National Institutes for Standards and

Technology, NIST), предприняла попытку создания стандарта для IDEF0 и IDEF1X. Эта попытка оказалась успешной и завершилась принятием в 1993 году стандарта правительства США, известного как FIPS для данных двух технологий IDEF0 и IDEF1X. В течение последующих лет этот стандарт продолжал активно развиваться и послужил основой для реализации в некоторых первых CASE средствах.

## 2.Нотация IDEF0

**IDEF0** используется для создания *функциональной модели*, отображающей структуру и функции системы, а также потоки информации и материальных объектов, преобразуемые этими функциями.

Методология IDEF0 незначительно отличается от классической схемы описания бизнес-процессов DFD. Основным отличием является классификация входов работы.

### Классификация входов и выходов работ

Стандарт предлагает следующую типизацию входов работ:

- **Вход.** Входит в работу слева и показывает информационные и материальные потоки, которые преобразуются в бизнес-процессе.
- **Управление.** Входит в работу сверху и показывает материальные и информационные потоки, которые не преобразуются в процессе, но нужны для его выполнения.
- **Механизм.** Входит в работу снизу и показывает людей, технические средства, информационные системы и т.п., при помощи которых бизнес процесс реализуется.
- **Результаты** выходят из блока справа

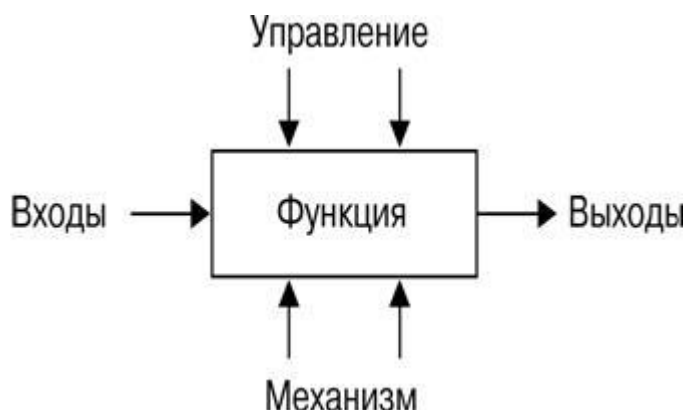


Рис. 1. Простейшая модель нотации

### Основные элементы диаграммы (Таблица 1):

Основу графического языка IDEF0, синтаксис и семантика которого определены с абсолютной строгостью, составляют блоки и соединяющие их стрелки, которые формируют иерархию детализируемых диаграмм.

Элемент	Графическое отображение и значение	Требования к оформлению
<b>Функциональный блок</b>	Изображается в виде прямоугольника. Представляют функции, определяемые как деятельность, процесс, операция, действие или преобразование.	1. Должен иметь уникальный идентификационный номер в правом нижнем углу; 2. Название должно быть в отглагольном наклонении.
<b>Интерфейсная дуга (стрелка, дуга)</b>	Изображается в виде однонаправленной стрелки. Представляют данные или материальные объекты, связанные с функциями.	1. Должна иметь уникальное наименование. 2. Наименование должно быть оборотом существительного. 3. Началом и концом дуги могут быть только функциональные блоки. 4. Источником может быть только выходная сторона блока, а приемником любая из трех оставшихся.

**Таблица 1. Основные элементы диаграммы**

IDEF0 — модель:

Модель включает следующие документы, которые ссылаются друг на друга:

- **Графические диаграммы** — главный компонент IDEF0-модели, который графически, с помощью блоков и стрелок и их соединений, отображает информацию о моделируемой системе. Блоки представляют основные функции. Эти функции могут быть разбиты (декомпозированы) на составные части и представлены в виде более подробных диаграмм. Процесс декомпозиции продолжается до тех пор, пока объект не будет описан на уровне детализации, необходимом для достижения целей конкретного проекта.
- **Текст;**
- **Глоссарий** — для каждого элемента диаграммы создается и поддерживается набор определений, ключевых слов, пояснений,

характеризующих объект, который представляет данный элемент. Этот набор называется глоссарием и является описанием сущности данного элемента. Глоссарий гармонично дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

## Принцип декомпозиции при построении модели бизнес процессов

### 1. Контекстная диаграмма: цель и точка зрения

Моделирование бизнес процесса начинается с контекстной диаграммы. Эта диаграмма называется А-0 (А минус ноль). На ней система представляется в виде одного блока и дуг, изображающих окружение системы. С помощью диаграммы можно увидеть взаимодействие моделируемой системы с внешней средой, все ее входы и выходы. Диаграмма А-0 устанавливает область моделирования и границы.

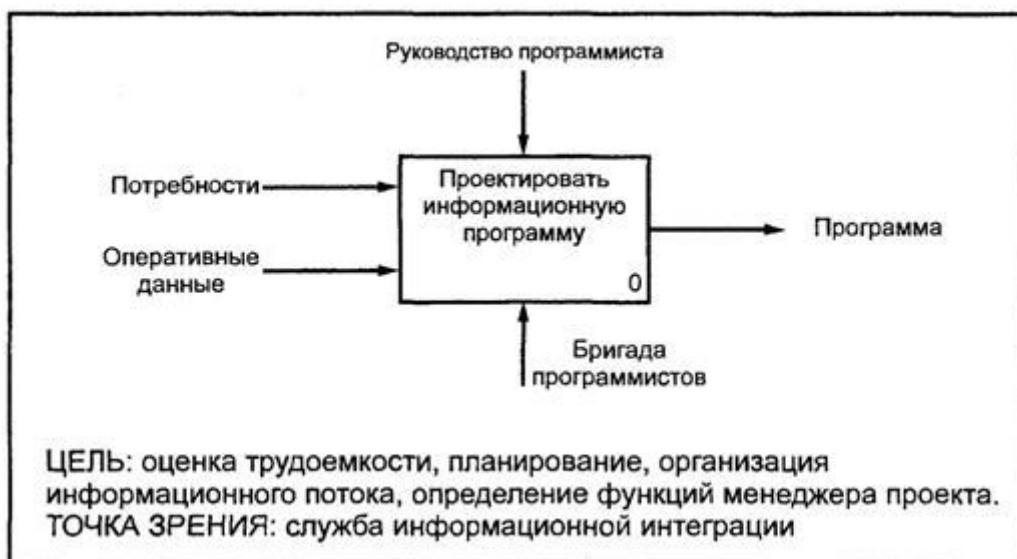


Рис. 2. Пример диаграммы

В пояснительном тексте к контекстной диаграмме должна быть указана цель построения диаграммы и зафиксирована точка зрения. Точка зрения определяет уровень детализации, направление развития модели и позволяет разгрузить модель. Так при моделировании можно отказаться от детализации и исследования отдельных элементов, не являющихся необходимыми, исходя из выбранной точки зрения на систему.

### 2. Детализация

Затем блок, который отображает всю систему, детализируется на другой диаграмме.

Далее каждая функция диаграммы может быть детализирована на дочерней. Каждая функция моделируется отдельным блоком. Каждый родительский блок подробно описывается дочерней диаграммой на более низком уровне. Так происходит до тех пор, пока не будет получена структура, позволяющая ответить на вопросы, сформулированные в цели моделирования.



Рис. 3. Детализация

Для достижения структурной целостности модели используются следующие правила:

- Все интерфейсные дуги, входящие в данный блок, или исходящие из него фиксируются на дочерней диаграмме.
- При нумерации блоков, цифра в правом нижнем углу прямоугольника указывает на уникальный порядковый номер самого блока на диаграмме, а обозначение под правым углом указывает на номер дочерней для этого блока диаграммы.

### Принцип туннелирования

Часто бывают случаи, когда отдельные стрелки не имеет смысла продолжать рассматривать в дочерних диаграммах ниже какого-то определенного уровня в иерархии, или наоборот — отдельные блоки не имеют практического смысла выше какого-то уровня. С другой стороны, иногда возникает необходимость избавиться от отдельных “концептуальных” стрелок и не детализировать их глубже некоторого уровня.



Для решения подобных задач в стандарте IDEF0 предусмотрено понятие туннелирования. Обозначение “туннеля” в виде двух круглых скобок вокруг начала стрелки обозначает, что эта стрелка не была унаследована от функционального родительского блока и появилась (из “туннеля”) только на этой диаграмме. В свою очередь, такое же обозначение вокруг конца стрелки в непосредственной близости от блока – приёмника означает тот факт, что в дочерней по отношению к этому блоку диаграмме эта стрелка отображаться и рассматриваться не будет.

### **Принцип ограничения сложности**

Для того, чтобы ограничить перегруженность моделей и сделать их удобными для восприятия, в стандарте приняты соответствующие ограничения сложности:

- ограничение количества функциональных блоков на диаграмме тремя-шестью. Верхний предел (шесть) заставляет разработчика использовать иерархии при описании сложных предметов, а нижний предел (три) гарантирует, что на соответствующей диаграмме достаточно деталей, чтобы оправдать ее создание;
- ограничение количества подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг четырьмя.

Разумеется, строго следовать этим ограничениям вовсе необязательно, однако, как показывает опыт, они являются весьма практичными в реальной работе.

### **Задания к выполнению:**

1. Разработать функциональные модели следующих предметных областей:
  - библиотечный каталог;
  - отдел кадров университета;
  - делопроизводство предприятия;
  - учет материальных ценностей (склад).

### **Контрольные вопросы:**

1. Понятие Case-средств и их назначение.
2. Назначение и сущность методологии IDEF0.

3. Направления IDEF0-моделирования.
4. Этапы жизненного цикла программных средств, для которых наиболее эффективно использование методологии IDEF0.
5. Достоинства методологии IDEF0.
6. Цель модели в IDEF0.
7. "Точка зрения" модели в IDEF0.
8. Правила представления работ на IDEF0-диаграмме.
9. Принцип ограничения сложности.
10. Разветвления дуг и правила их обозначения на IDEF0-диаграмме.

### 3.Методология IDEF3

Методология IDEF3 является одним из стандартов семейства IDEF и довольно широко используется при декомпозиции моделей IDEF0 для моделирования процессов более низкого уровня, поскольку с его помощью можно смоделировать технологические процессы, происходящие на предприятии, т.е. описать возможные сценарии реализации процессов, в рамках которых происходит последовательное изменение свойств объекта. Данная методология позволяет показывать возможные разветвления в процессе. Например, когда результат одного действия может инициировать запуск нескольких действий или наоборот, чтобы начать какое-то действие, необходимо завершить несколько предыдущих действий.

Модели IDEF3 можно отнести к классу WFD-диаграмм, поскольку с их помощью также описывается взаимосвязанная последовательность действий, которые осуществляются в рамках реализации процесса.

В рамках стандарта IDEF3 выделяют два типа диаграмм, позволяющих описать процесс с разных точек зрения:

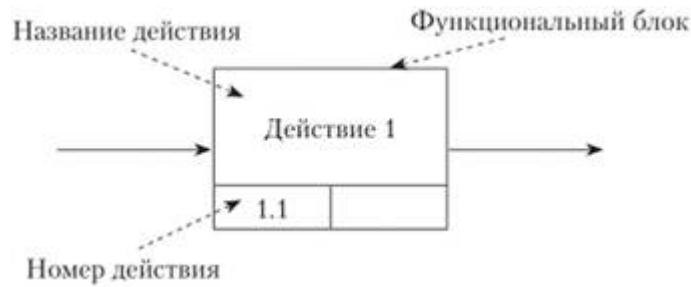
- • диаграмма описания последовательности этапов процесса (Process Flow Description Diagrams — PFDD), с помощью которой моделируется последовательность действий, реализуемых в рамках бизнес-процесса;
- • диаграмма состояния и трансформации объекта в процессе (Object State Transition Network — OSTN), с помощью которой описываются изменения, происходящие с объектом в ходе его обработки.

Для описания и моделирования бизнес-процессов, где основной задачей стоит описание последовательностей действий, которые необходимо выполнить для достижения поставленных целей, большой интерес представляют диаграммы типа PFDD. Рассмотрим его подробнее.

Основными элементами диаграммы PFDD IDEF3 (далее — IDEF3) являются:

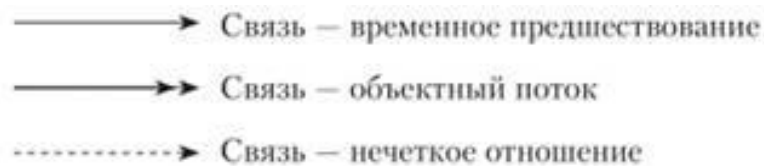
- • функциональный элемент;
- • стрелка;
- • перекресток.

Функциональный элемент (элемент поведения, единица работы) используется для обозначения действия, работы или события. Он отражается в виде прямоугольника, в центре которого указывается название действия (глагол или отглагольное существительное). Внизу блока указывается номер действия с учетом номера родительской диаграммы (рис. 4)



**Рис. 4.** Структура функционального элемента в IDEF3

**Стрелка (линия)** используется для отражения последовательности выполнения работ (действий) и связей между ними. Все стрелки показывают движение в одну сторону: слева направо, таким образом, визуально соблюдая идею демонстрации последовательного выполнения операций процесса. Они могут выходить и входить с любой стороны блока, но предпочтение лучше отдавать их горизонтальному расположению. Существуют три типа стрелок (рис. 5.): временное предшествование, объектный поток, нечеткое отношение.



**Рис. 5.** Типы стрелок в нотации IDEF3

Стрелка типа **"Временное предшествование"** показывает, что действие, из которого она выходит, должно завершиться до того, как начнется действие, в которое она входит. Результат исходного действия не обязательно является инициатором для действия, куда входит стрелка. Главное значение данной стрелки — показать временную связь между действиями, т.е. показать, что одно действие не может начаться до того, пока предыдущее не закончится, независимо от результата его завершения. Такая связь обозначается простой стрелкой.

Стрелка типа **"Объектный поток"** показывает, что результат действия, из которого она выходит, является инициатором действия, в которое оно входит. Соответственно действие, в которое входит стрелка, не может начаться до тех пор, пока не закончится действие, из которого стрелка выходит. Такая связь обозначается стрелкой с двойным наконечником. В названии стрелки должно быть приведено название объекта, который передается от одной операции к другой.

Стрелка типа **"Нечеткое отношение"** показывает, что тип связи между двумя действиями задается индивидуально, может иметь переменчивый или уникальный характер. Такая связь обозначается пунктирной стрелкой.

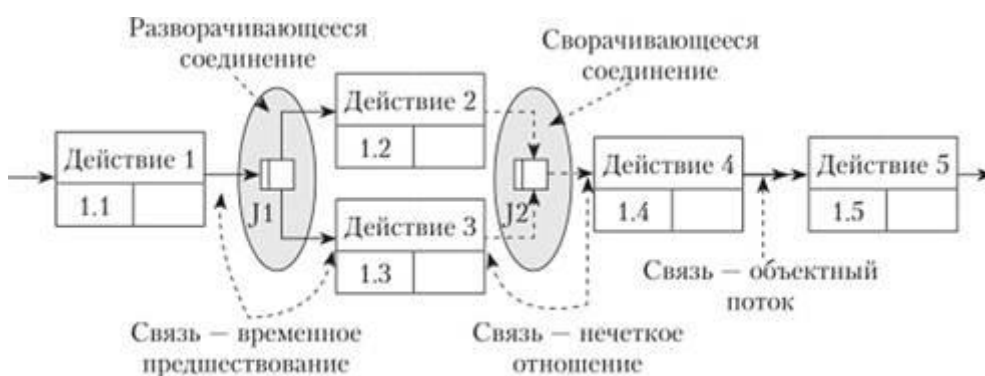
специальных требований по ее наименованию нет. Такое изображение связей используется, когда нельзя применить связи, типа "Временное предшествование" и "Объектный поток".

**Перекресток (условные символы ветвления)** используется для отражения логики движения потоков между функциональными элементами (операциями). Перекресток позволяет указать события, которые могут или должны произойти для того, чтобы началось выполнение следующего действия. На диаграмме IDEF3 перекресток представляет собой прямоугольник с индикатором "J" и номером данного перекрестка на диаграмме (рис. 6.). Существуют перекрестки, используемые для отражения слияния стрелок, и перекрестки, используемые для отражения разветвления стрелок. Стоит отметить, что один перекресток не может одновременно использоваться для слияния и для разветвления. В методологии IDEF3 выделяют: **разворачивающиеся соединения**, используемые для отражения связей, где завершение одного процесса инициирует запуск нескольких других процессов: **сворачивающиеся соединения**, используемые для отражения связей, где завершение нескольких процессов приводит к запуску следующего одного процесса.

Разворачивающиеся и сворачивающиеся соединения могут быть также нескольких типов:

- "и" (обозначается квадратом с символом "&");
- "исключающее "или"" (обозначается квадратом с символом "X");
- "или" (обозначается квадратом с символом "O").

На рис. 6 приведен образец построения IDEF3-диаграммы.



**Рис. 6.** Образец диаграммы в нотации IDEF3

Соединение типа "и" используется для описания ситуаций, когда:

- только после завершения нескольких действий может наступить следующее действие;

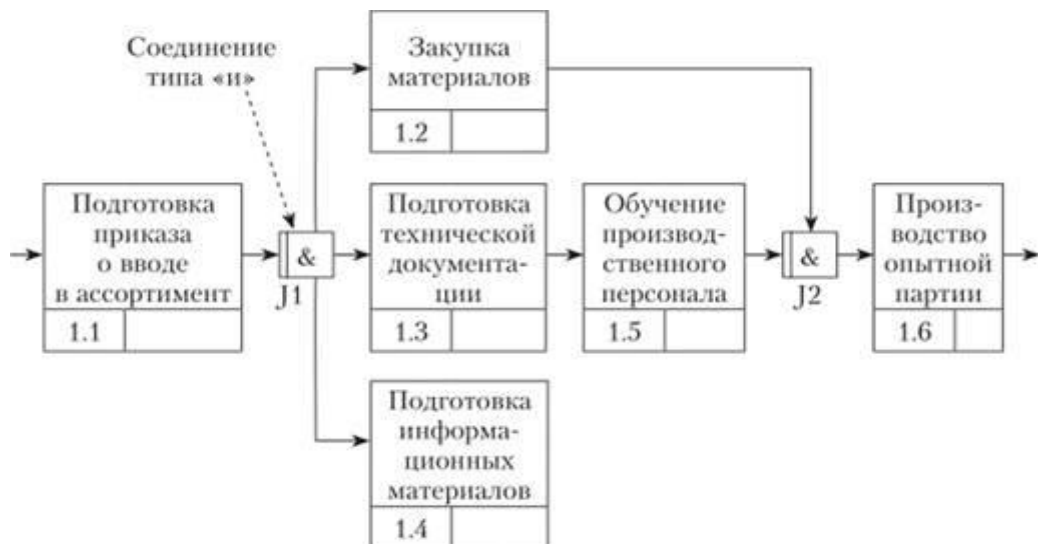
- после завершения действия одновременно запускаются несколько следующих действий.

Следует учитывать, что если соединение "и" инициирует выполнение последнего действия, то все действия, которые присоединяются к сворачиваемому соединению типа "и" должны быть выполнены полностью.

Например, процесс "Подготовка к продаже нового изделия" состоит из следующих подпроцессов (рис. 7):

- 1.1. Подготовка приказа о вводе в ассортимент нового продукта.
- 1.2. Закупка материалов для производства изделия.
- 1.3. Подготовка технической документации по изготовлению нового изделия.
- 1.4. Подготовка информационных материалов для продвижения и продажи.
- 1.5. Обучение производственного персонала изготовлению нового изделия.
- 1.6. Производство опытной партии нового изделия.

Процессы "Закупка материалов для производства изделия", "Подготовка информационных материалов для продвижения и продажи" и "Подготовка технической документации по изготовлению изделия" начинаются сразу после того, как выпущен приказ о вводе в ассортимент нового продукта. Процесс "Производство опытной партии нового изделия" может начаться только после того, как обучен производственный персонал и закуплен материал для производства.



**Рис.7.** IDEF3-диаграмма процесса "Подготовка к продаже нового изделия"

Соединение типа "исключающее "или"" используется для описания ситуаций, когда:

- после завершения одного действия может начаться только одно из следующих действий;
- следующее действие может начаться после завершения только одного из предыдущих действий.

Например, соединение "исключающее "или"" используется для того, чтобы показать, что результатом согласования проекта договора может быть: а) проект договора согласован; б) по проекту договора есть замечания и он отправлен на доработку (рис. 8). В первом случае, если он согласован, то осуществляется следующее действие — подписание договора. Во втором случае, когда по нему есть замечания, осуществляется его доработка. Здесь, "исключающее "или"" показывает, что в зависимости от результата выполнения первого действия потом будет выполняться второе или третье действие.

При использовании такого типа соединения целесообразно подписывать стрелки или делать комментарии к ним, показывая в каком случае, какое действие будет выполняться.

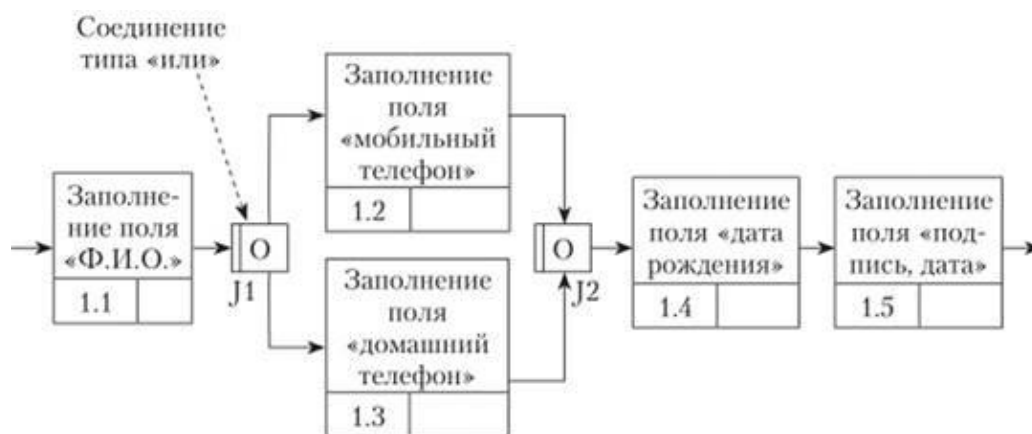


**Рис. 8.** Фрагмент IDEF3-диаграммы процесса "Управление договором"

Соединение типа "или" используется для описания ситуаций, при которых:

- после завершения одного или нескольких предшествующих действий может наступить следующее действие;
- после завершения одного действия может начаться одно или несколько следующих действий.

Примером использования такого типа соединения может служить фрагмент процесса заполнения анкеты, представленный на рис. 9.



**Рис. 9.** Фрагмент IDEF3-диаграммы процесса "Заполнение анкеты"

На рис. 9 показан пример использования соединения типа "или", где после действия "Заполнение поля «Ф.И.О.»" может быть выполнено действие "Заполнение поля «мобильный телефон»" или действие "Заполнение поля «домашний телефон»" либо оба эти действия. Одно из них точно должно быть выполнено.

Таким образом, можно выделить пять типов перекрестков. Каждый из них имеет свое обозначение. В табл. 4 приведено краткое описание всех типов перекрестков.

Обозначение	Наименование	Смысл в случае слияния стрелок	Смысл в случае разветвления стрелок
&	Асинхронное "и"	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
&	Синхронное "и"	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
O	Асинхронное "или"	Один или несколько предшествующих процессов должны быть завершены	Один или нескольких следующих процессов должны быть запущены
O	Синхронное "или"	Один или несколько	Один или несколько



		предшествующих процессов завершаются одновременно	следующих процессов запускаются одновременно
<b>X</b>	Исключающее "или"	Только один предшествующий процесс завершен	Только один следующий процесс запускается

**Таблица 2.** Типы перекрестков.

В приведенных выше примерах IDEF3-диаграмм используются асинхронные типы перекрестков, поскольку на практике они встречаются чаще, нежели синхронные.

Процессы, описанные с помощью IDEF3-диаграмм, могут быть также декомпозированы для более детального анализа.

Модели в нотации IDEF0 могут быть декомпозированы в виде IDEF0- и IDEF3-диаграмм, а модели IDEF3 могут быть декомпозированы только в виде IDEF3-диаграмм.

Используя диаграмму процесса в нотации IDEF0 в качестве родительской диаграммы, можно построить дочерние для ее функциональных блоков модели в нотации IDEF3. При нумерации функциональных элементов IDEF3-диаграмм необходимо учитывать номера функциональных блоков родительской IDEF0-диаграммы. Здесь работает правило декомпозиции методологии SADT.

Однако стоит учитывать, что модели IDEF3 могут быть декомпозированы только в виде IDEF3-диаграмм.

### **Задания к выполнению:**

- 1) Составить диаграмму декомпозиции на основе функциональных блоков:
  - 1) Проверка продукции на годность.
  - 2) Принять продукцию.
  - 3) Вернуть поставщику.
  - 4) Продукция проверена.
- 2) Составить диаграмму декомпозиции на основе функциональных блоков:

- 1) Покрасить деталь.
  - 2) Сушить деталь.
  - 3) Тестировать деталь.
  - 4) Окрасить заново.
  - 5) Отправить в следующий цех.
- 3) Составить диаграмму:
- 1) Заказы на настольные компьютеры
  - 2) Проверка наличия необходимых для сборки комплектующих
  - 3) Заказ комплектующих со склада
  - 4) Список необходимых комплектующих
  - 5) Подготовка комплектующих
  - 6) Необходимые комплектующие
  - 7) Установка материнской платы и процессора
  - 8) Установка ОЗУ
  - 9) Установка винчестера
  - 10) Установка DVD
  - 11) Установка ТВ-тюнера
  - 12) Установка кард-ридера
  - 13) Установка ОС
  - 14) Установка дополнительного ПО
  - 15) Составление отчета о результатах сборки

### **Контрольные вопросы:**

1. Что описывает диаграмма IDEF3?
2. Перечислите составные элементы диаграмм IDEF3.
3. Что показывают связи в диаграммах IDEF3?
4. Перечислите типы стрелок в диаграммах IDEF3.
5. Что называется перекрестком?
6. Назовите типы перекрестков.

## **4.Методология ARIS**

### **Методология ARIS**

Одной из современных методологий бизнес-моделирования, получившей широкое распространение в России является методология ARIS, которая расшифровывается как Architecture of Integrated Information Systems - проектирование интегрированных информационных систем.

Методология ARIS на данный момент времени является наиболее объемной и содержит около 100 различных бизнес-моделей, используемых для описания, анализа и оптимизации различных аспектов деятельности организации. Часть моделей методологии ARIS используются в настройке модуля интегрированной информационной системы SAP/R3, который применяется при внедрении системы и ее настройке на деятельности компании. В виду большого количества бизнес-моделей методология ARIS делит их на четыре группы (рис. 10):

#### **Группа "Оргструктура".**

Состоит из моделей с помощью которых описывается организационная структура компании, а также другие элементы внутренней инфраструктуры организации.

#### **Группа "Функции".**

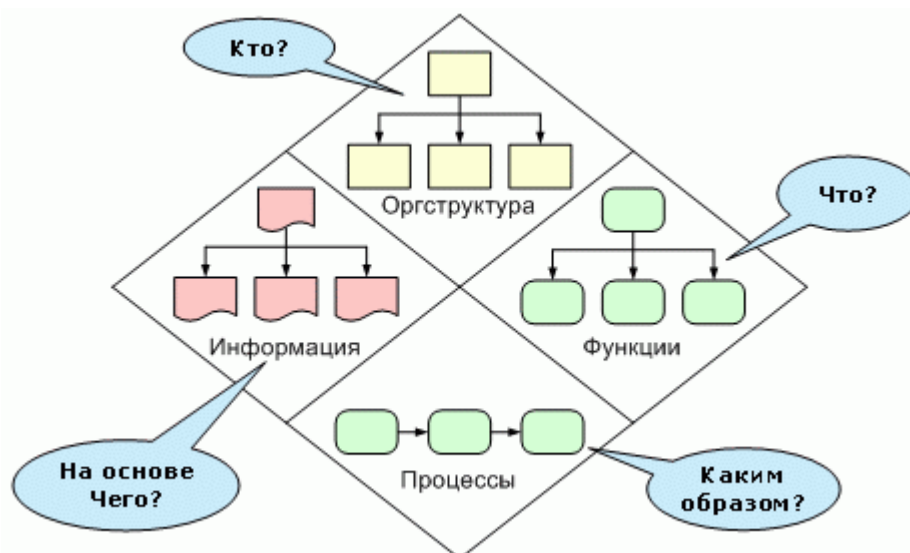
Состоит из моделей, используемых для описания стратегических целей компании, функций и прочих элементов функциональной деятельности организации.

#### **Группа. "Информация".**

Состоит из моделей с помощью которых описывается информация, используемая в деятельности организации.

#### **Группа "Процессы".**

Состоит из моделей, используемых для описания бизнес-процессов, а также различных взаимосвязей между структурой, функциями и информацией.



**Рис. 10.** Группы моделей методологии ARIS.

Большим преимуществом методологии ARIS является эргономичность и высокая степень визуализации бизнес-моделей, что делает данную методологию удобной и доступной в использовании всеми сотрудниками компании, начиная от топ-менеджеров и заканчивая рядовыми сотрудниками. В методологии ARIS смысловое значение имеет цвет, что повышает восприимчивость и читабельность схем бизнес-моделей. Например, структурные подразделения по умолчанию изображаются желтым цветом, бизнес-процессы и операции - зеленым. Помимо большего количества моделей по сравнению с другими методологиями, методология ARIS имеет наибольшее количество различных объектов, используемых при построении бизнес-моделей, что увеличивает их аналитичность. Например, материальные и информационные потоки на процессных схемах обозначаются разными по форме и цвету объектами, что позволяет быстро определить тип потока.

Несмотря на большее количество моделей в методологии ARIS в проектах по описанию и оптимизации деятельности в общем случае их используется не более десяти. Методология ARIS позиционирует себя как конструктор, из которого под конкретный проект в зависимости от его целей и задач разрабатывается локальная методология, состоящая из небольшого количества требуемых бизнес-моделей и объектов. В общем случае практика показала, что в проектах наиболее часто используются модели, приведенные в таблице 3.

№	Название модели		Описание и предназначение модели
	Английский вариант	Русский вариант	
1	OD - Objective diagram.	Диаграмма целей.	Модель описывает стратегические цели компании и их взаимосвязь

			с другими элементами организации
2	PST - Product/Service tree.	Дерево продуктов и услуг.	Модель описывает продукты и услуги, производимые компанией и их взаимосвязь с другими элементами организации.
3	FT - Function tree.	Дерево функций.	Модель описывает функции, выполняемые в компании и их иерархию.
4	FAD - Function allocation diagram.	Диаграмма окружения процесса.	Процессная модель описывает окружение бизнес-процесса.
5	VACD - Value added chain diagram.	Диаграмма цепочки добавленной стоимости.	Процессная модель - прототип классического стандарта DFD. Применяется для описания бизнес-процессов верхнего уровня.
6	PSM - Process selection matrix.	Матрица выбора процесса.	Процессная модель - прототип классического стандарта DFD. Является альтернативой модели VACD и применяется для описания бизнес-процессов верхнего уровня.
7	eEPC - Extended event driven Process Chain.	Расширенная цепочка процессов, управляемая событиями.	Процессная модель прототип классического стандарта WFD. Применяется для описания бизнес-процессов нижнего уровня.
8	OC - Organizational chart.	Модель организационной структуры.	Модель описывает организационную структуру компании.
9	ASTD - Application system type diagram.	Диаграмма типов информационных систем.	Модель описывает структуру информационных систем, используемых в компании.

**Таблица 3. «Методологии ARIS»**

**Модель "Диаграмма целей" - OD** применяется для описания стратегических целей компании, их иерархической упорядоченности, а также связей целей с продуктами и услугами, производимыми компанией и бизнес-процессами, поддерживающими их производство (рис. 11)

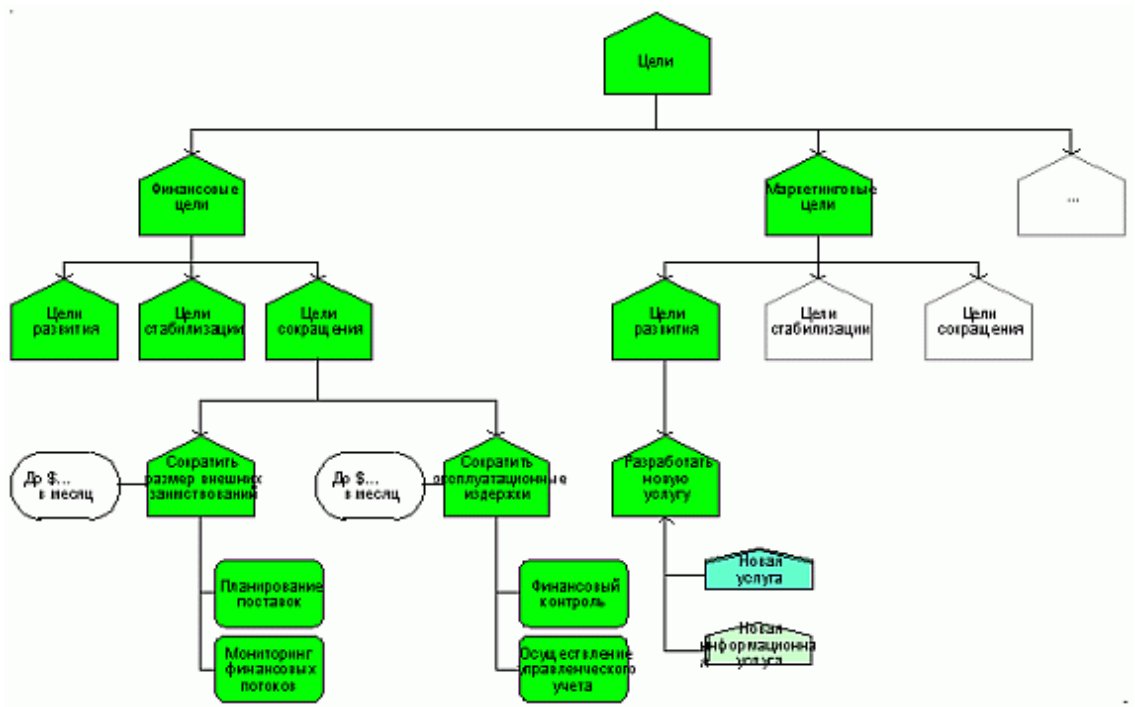


Рис. 11. Модель "Диаграмма целей" - OD.

**Модель "Дерево продуктов и услуг" - PST** применяется для описания продуктов и услуг, производимых в компании, а также и связи со стратегическими целями компании, бизнес-процессами, поддерживающими их производство (рис. 12).

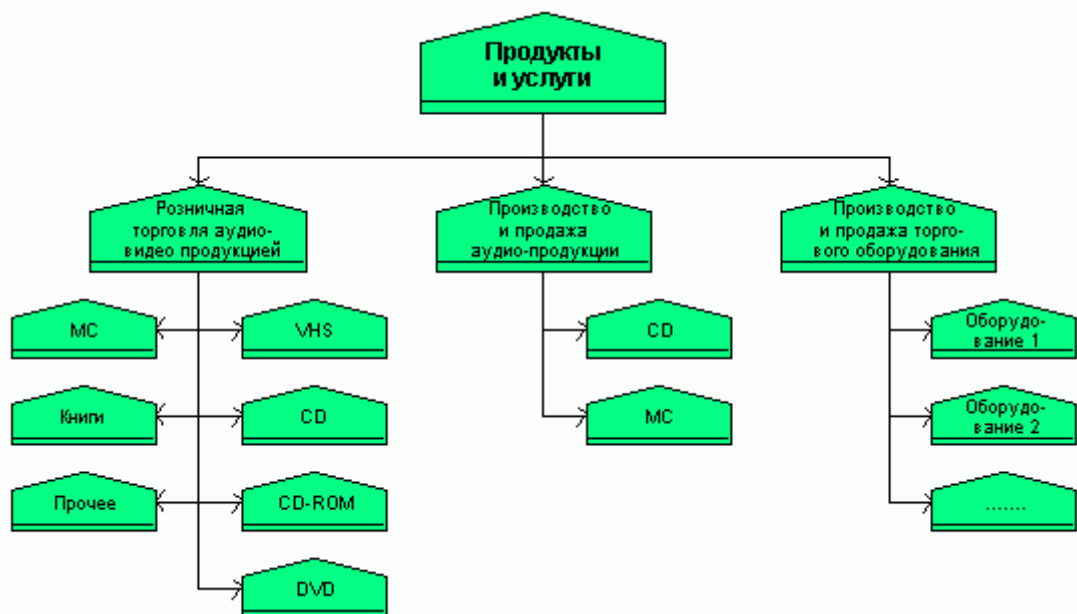


Рис. 12. Модель "Дерево продуктов и услуг" - PST.

**Модель "Дерево функций" - FT** описывает функции, выполняемые в компании и их иерархию. Данная модель часто применяется для построения дерева бизнес-процессов компании (рис. 13).

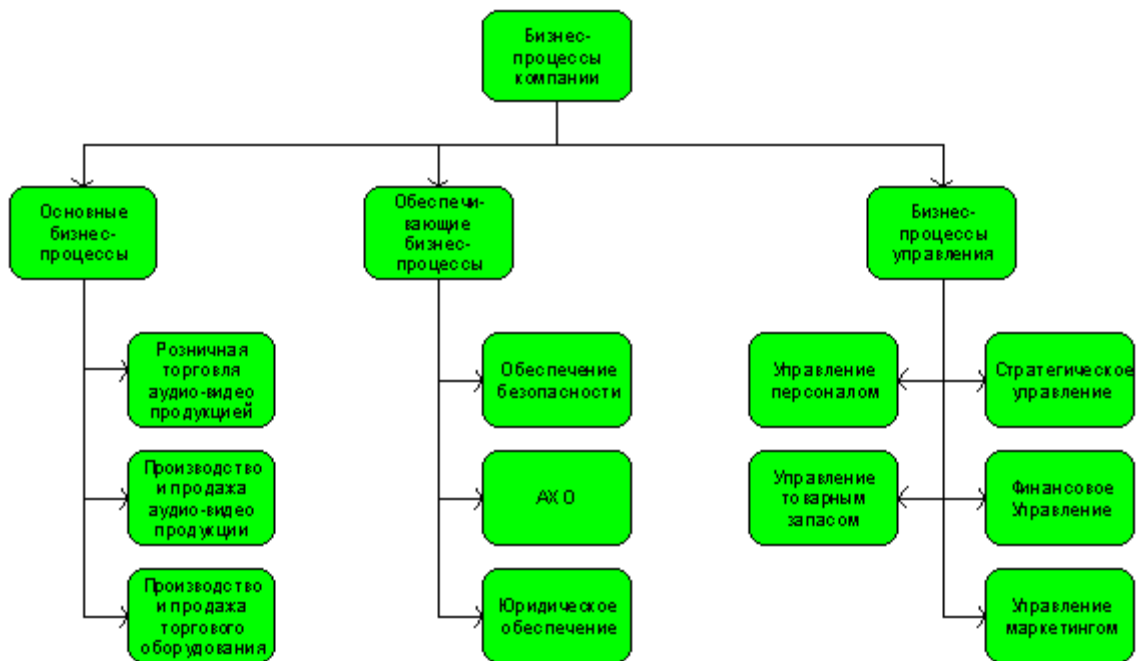


Рис. 13. Модель "Дерево функций" - FT.

Модель "Диаграмма окружения процесса" - FAD позволяет описать окружение или границы бизнес-процесса, показывая его входы, выходы, поставщиков и клиентов (рис. 14).



Рис. 14. Модель "Диаграмма окружения процесса" - FAD.

Модель "Диаграмма цепочки добавленной стоимости" - VACD является прототипом классического DFD-стандарта и используется для описания бизнес-процессов верхнего уровня. Дополнительным отличием данной и других процессных моделей является то, что информационные и материальные потоки на схеме VACD изображаются не стрелками, а объектами. При этом для каждого типа потока используется свой объект. На модели VACD методологии ARIS в отличие от классического подхода также используется логические связи между работами, которые позволяют отобразить логическую последовательность выполнения работ. В качестве одного из вариантов логической последовательности может выступать

временная последовательность выполнения работ, что характерно для классического подхода WFD. (рис. 15).



Рис. 15. Модель "Диаграмма цепочки добавленной стоимости" - VACD.

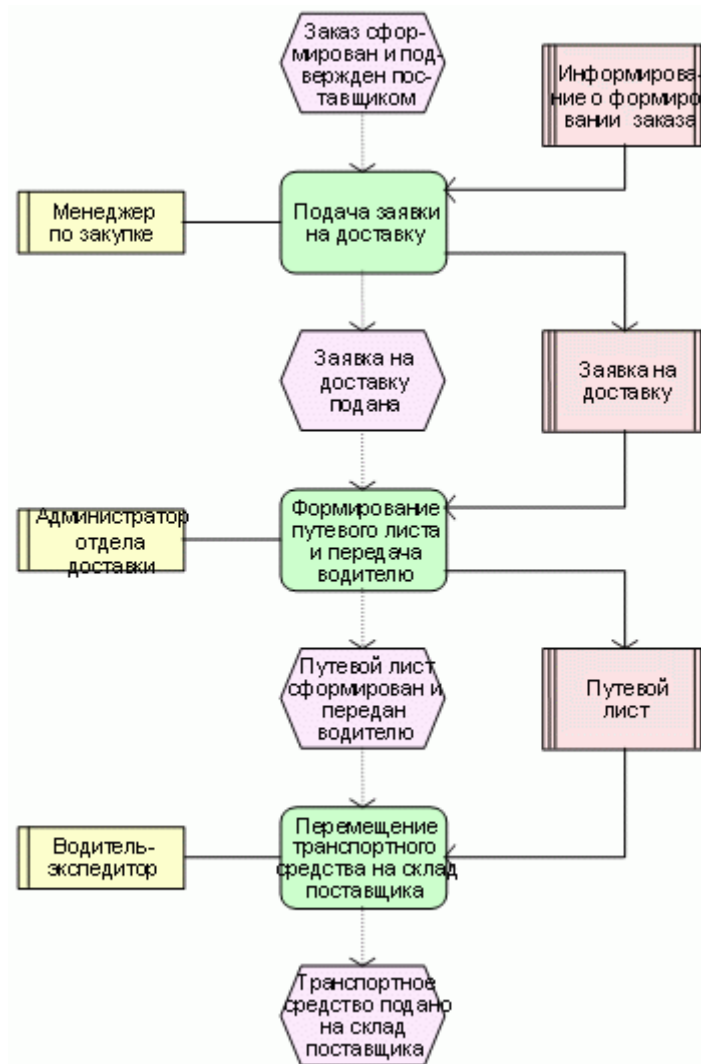
**Модель "Матрица выбора процесса" - PSM** является прототипом классического DFD-стандарта и используется как альтернатива для модели VACD. Матрица выбора процессов по отношению к диаграмме цепочки добавленной стоимости является с одной стороны более упрощенным вариантом описания процесса, с другой стороны данная модель содержит дополнительные объекты, позволяющие показать другие аспекты бизнес-процесса. Простота матрицы выбора бизнес-процессов связана с тем, что на данной модели не показываются информационные и материальные потоки. Что касается других аспектов, то данная модель позволяет на одной схеме компактно и наглядно показать различные варианты выполнения бизнес-процесса, который описывается. Соответственно матрицу выбора процессов целесообразно применять вместо диаграммы цепочки добавленной стоимости в случаях, когда описываемый бизнес-процесс имеет несколько вариантов исполнения, каждый из которых ложится базовую схему. Пример применения матрицы выбора процессов для описания деятельности компании "Эврика", имеющий функциональную организационную структуру показан на рис. 16.



Рис. 16. Модель "Матрица выбора процессов" - PSM.



**Модель "Extended event driven Process Chain" - eEPC** является прототипом классического WFD-стандарта и используется для описания бизнес-процессов нижнего уровня. Дополнительным отличием eEPC-модели от классической WFD-схемы является наличие на модели объекта, который называется событием. С помощью событий изображается факт, время или событие инициирующие начало выполнения работ процесса, а также факт или время их завершения (рис. 17).



**Рис. 17.** Модель "Расширенная цепочка процессов, управляемая событиями" - eEPC.

**Модель "Организационная структура" - ОС** используется для описания организационной структуры компании. На данной модели изображаются структурные подразделения, группы, должности, роли и прочие элементы организационной структуры и связи между ними (рис. 18).

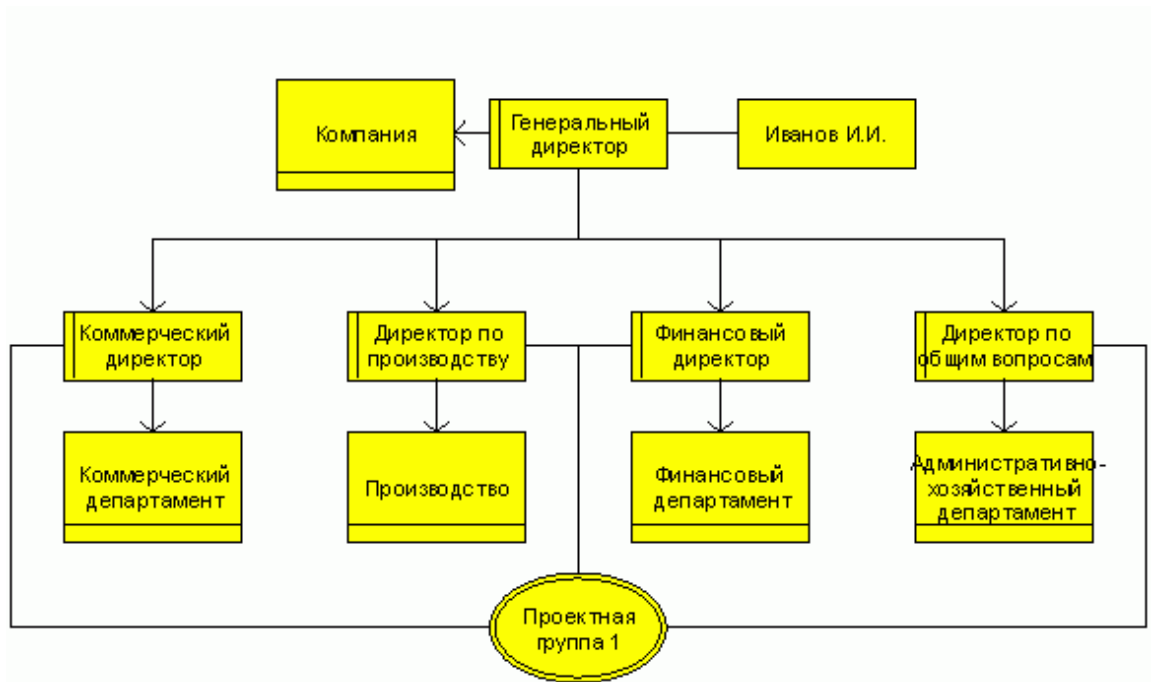


Рис. 18. Модель "Организационная структура" - ОС.

Модель "Диаграмма типов информационных систем" - ASTD используется для описания структуры информационных систем, используемых в компании. На данной модели показываются типы и модули информационных систем, программные продукты, взаимосвязь между ними и бизнес-процессами организации, которые они автоматизируют (рис. 19).

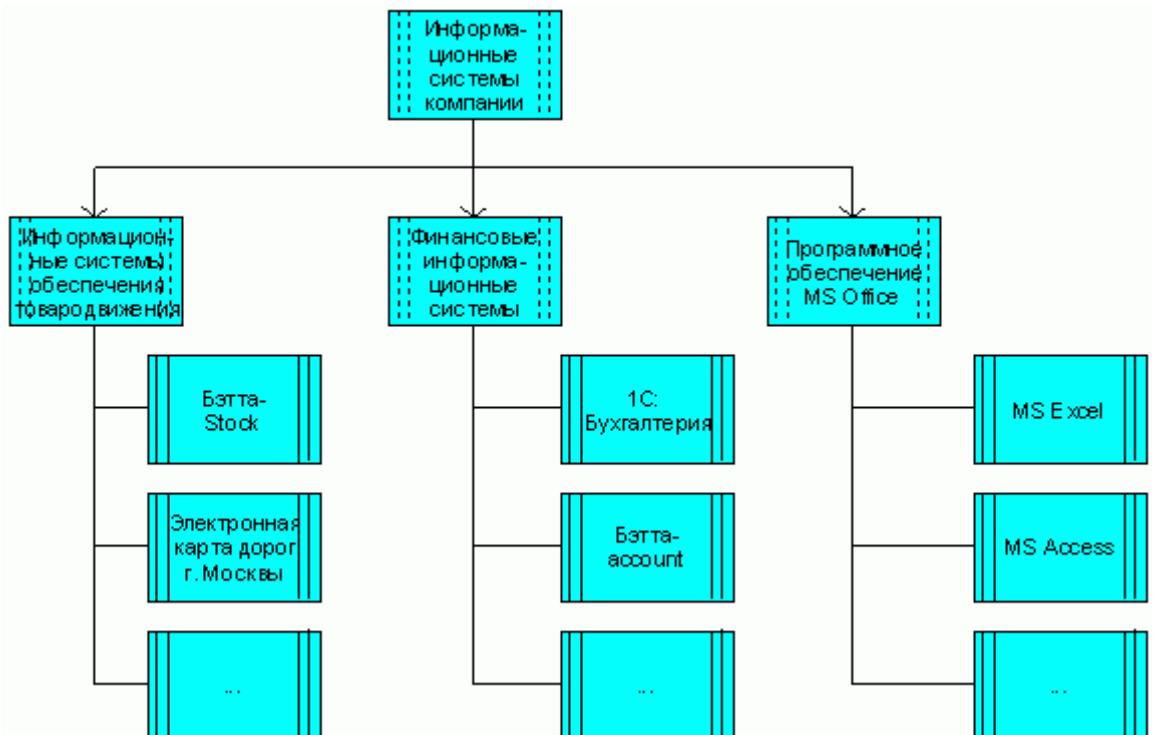
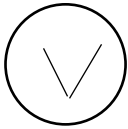
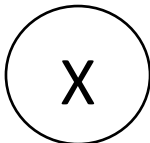


Рис. 19. Модель "Диаграмма типов информационных систем" - ASTD.

## 5. Описание нотации ARIS eEPC

Нотация ARIS eEPC расшифровывается следующим образом – extended Event Driven Process Chain – расширенная нотация описания цепочки процесса, управляемого событиями. Нотация разработана специалистами компании IDS Scheer AG (Германия), в частности профессором Шеером. В таблице 2 приводятся основные используемые в рамках нотации объекты.

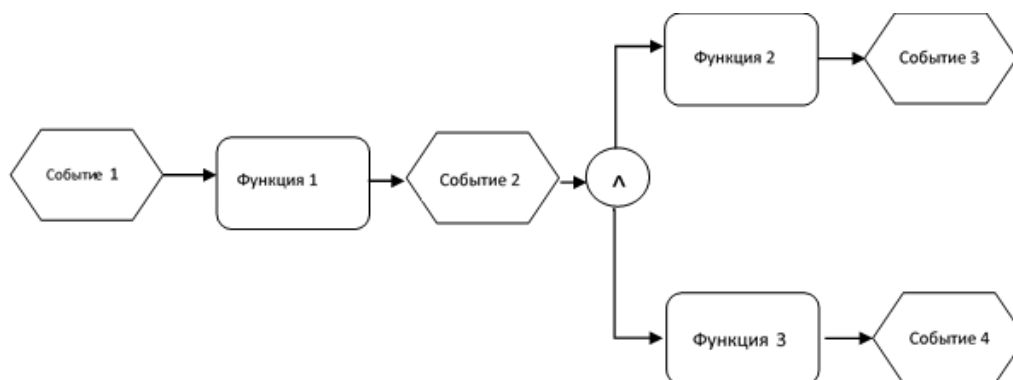
№	Наименование	Описание	Графическое представление
1	Функция	Объект «Функция» служит для описания функций (процедур, работ), выполняемых подразделениями/сотрудниками предприятия	
2	Событие	Объект «Событие» служит для описания реальных состояний системы, влияющих и управляющих выполнением функций	
3	Организационная единица	Объект, отражающий различные организационные звенья предприятия (например, управление или отдел)	
4	Документ	Объект, отражающий реальные носители информации, например, бумажный документ	
5	Прикладная система	Объект отражает реальную прикладную систему, используемую в рамках технологии выполнения функции	
6	Кластер информации	Объект характеризует данные, как набор сущностей и связей между ними. Используется для создания модели данных	
7	Стрелка связи между объектами	Объект описывает тип отношения между другими объектами, например – активацию выполнения функции некоторым событием	
8	Логическое «И»	Логический оператор, определяющий связи между	

		событиями и функциями в рамках процесса. Позволяет описать ветвление процесса	
9	Логическое «ИЛИ»	Логический оператор, определяющий связь между событиями и функциями в рамках процесса. Позволяет описать ветвление процесса	
10	Логическое исключаяющее «ИЛИ»	Логический оператор, определяющий связи между событиями и функциями в рамках процесса. Позволяет описать ветвление процесса.	

**Таблица 4.** «Основные объекты методологии ARIS»

Помимо указанных в Таблице 4 основных объектов, при построении диаграммы eEPC могут быть использованы многие другие объекты.

Применение большого числа различных объектов, связанных различными типами связей значительно увеличивает размер модели и делает ее плохо читаемой. Для понимания смысла нотации eEPC достаточно рассмотреть основные используемые типы объектов и связей. На следующем рисунке представлена простейшая модель eEPC, описывающая фрагмент бизнес-процесса предприятия.



**Рис. 20** «Фрагмент бизнес процесса предприятия»

На рисунке 20 видно, что связи между объектами имеют определенный смысл и отражают последовательность выполнения функций в рамках процесса. Стрелка, соединяющая Событие 1 и Функцию 1 “активирует” или инициирует выполнение Функции 1. Функция 1 “создает” Событие 2, за которым следует символ логического И, “запускающий” выполнение

Функций 2 и 3. Нотация eEPC построена на определенных семантических правилах описания:

- каждая функция должна быть инициирована событием и должна завершаться событием;
- в каждую функцию не может входить более одной стрелки, “запускающей” выполнение функции, и выходить не более одной стрелки, описывающей завершение выполнения функции.

На рисунке 21 показано применение различных объектов ARIS при создании модели бизнес-процесса.

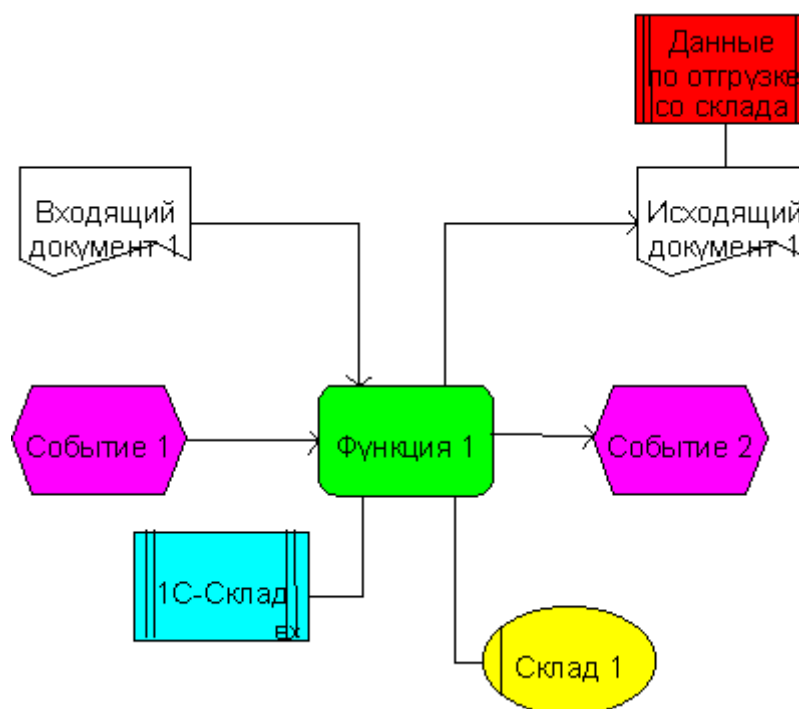


Рис. 21. «Применение объектов при создании модели процесса»

Из рисунка 21 видно, что бизнес-процесс в нотации eEPC представляет собой последовательность процедур, расположенных в порядке их выполнения. Следует отметить, что реальная длительность выполнения процедур в eEPC визуально отражена быть не может. Это приводит к тому, что при создании моделей возможны ситуации, когда на одного исполнителя будет возложено выполнение двух задач одновременно. Используемые при построении модели символы логики позволяют отразить ветвление и слияние бизнес-процесса.

Таким образом, при помощи нотации eEPC ARIS можно описывать бизнес-процесс в виде потока последовательно выполняемых работ (процедур,

функций). Пример модели, сформированной с использованием ARIS eEPC, показан на рисунке 22.

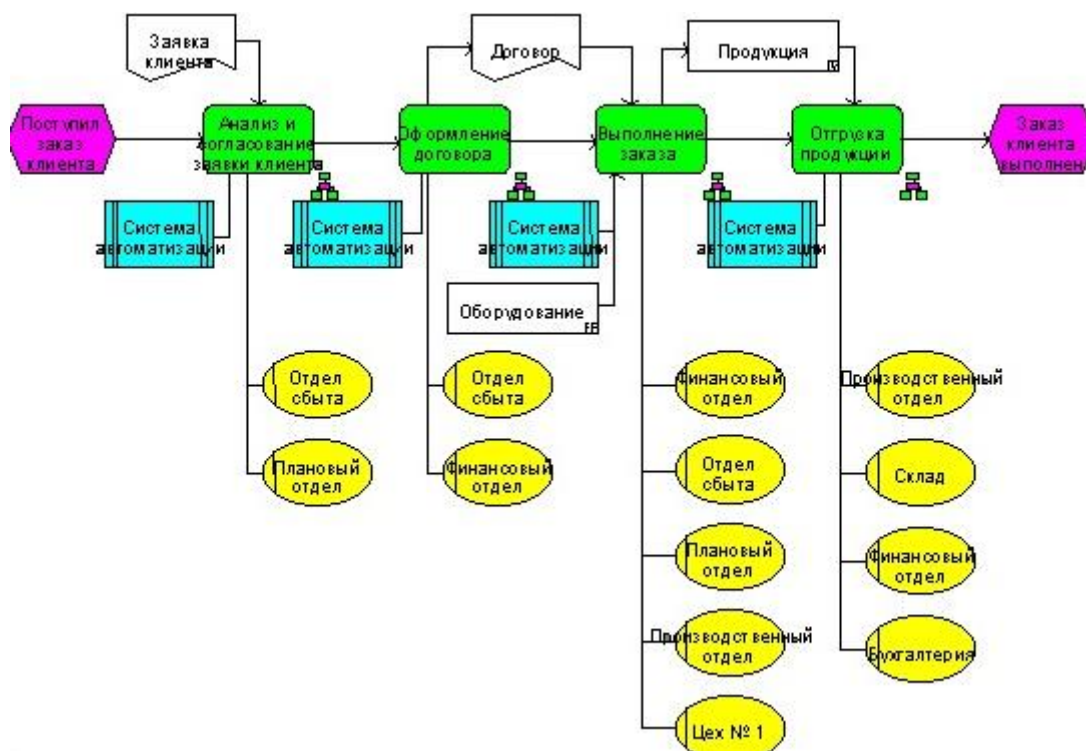


Рис. 22. Модель методологии ARIS

### Контрольные вопросы:

1. Основные используемые в рамках нотации объекты. Их описание.
2. На каких семантических правилах построена нотация?
3. На какие группы ARIS делит бизнес-модели? Дайте их определение.
4. Преимущества методологии ARIS.
5. Как система ARIS себя позиционирует?

### Задания к выполнению:

1. Построить Древо функций:
  - 1) Произвести продукцию
  - 2) Закупать материалы
  - 3) Осуществлять производство
  - 4) Продавать продукцию
  - 5) Планировать потребности
  - 6) Выявлять поставщиков
  - 7) Оплачивать и получать материалы

- 8) Планировать производство
  - 9) Управлять производством
  - 10) Получать и обрабатывать заказ
  - 11) Принять к исполнению
  - 12) Отследить выполнение заказа
2. Построить диаграмму
- 1) Заказ на производство
  - 2) Отследить заказ на производство
  - 3) Управлять производством
  - 4) Появилась необходимость во внешней детали
  - 5) Закупить деталь
  - 6) Отдел ИТ
  - 7) Сведения о поставщиках
  - 8) Внешняя деталь получена
  - 9) Заказ клиента обработан
  - 10) Изделие создано
  - 11) Отгрузить деталь
  - 12) Сопроводительные документы
  - 13) Заказ клиента обработан

## 6. Нотация DFD

Стандарт описания бизнес-процессов DFD — Data Flow Diagram переводится как диаграмма потоков данных и используется для описания процессов верхнего уровня и для описания реально существующих в организации потоков данных.


### Использование и особенности DFD диаграмм

Созданные модели потоков Данных организации могут быть использованы при решении таких задач, как:

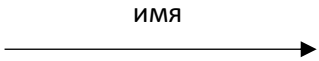


1. Определение существующих хранилищ данных (текстовые документы, файлы, Система управления базой данных — СУБД);
2. определение и анализ данных, необходимых для выполнения каждой функции процесса;
3. подготовка к созданию модели структуры данных организации, так называемая ERD-модель (IDEF1X);
4. выделение основных и вспомогательных бизнес-процессов организации.

Диаграммы потоков данных показывают, как каждый процесс преобразует свои входные данные в выходные, и выявляют отношения между этими процессами. DFD представляет моделируемую систему как сеть связанных работ.

При построении DFD-схемы бизнес-процесса **нужно помнить, что данная схема показывает потоки материальных и информационных потоков и ни в коем случае не говорит о временной последовательности работ**, хотя в большинстве случаев временная последовательность работ и совпадает с направлением движения потоков в бизнес-процессе.

Элемент	Описание	Форма представления
Функция	Работа. Функция или последовательность действий, которые нужно предпринять, чтобы данные были обработаны. Это может быть создание заказа, регистрация клиента и т.д. В названиях процессов принято использовать глаголы.	



Поток данных	В нотации отображается в виде стрелок, которые показывают, какая информация входит, а какая исходит из того или иного блока на диаграмме.	
Хранилище данных	Внутреннее хранилище данных для процессов в системе.	
Внешняя сущность	Это любые объекты, которые не входят в саму систему, но являются для нее источником информации либо получателями какой-либо информации из системы после обработки данных. Это может быть человек, внешняя система, какие-либо носители информации и хранилища данных.	

**Таблица 5.** Основные элементы нотации DFD

Требования к оформлению функций:

Каждая функция должна иметь идентификатор;

Названия работы нужно формулировать согласно следующему формуле:

Название работы = Действие + Объект, над которым действие осуществляется

Например, если эта работа связана с действием по продаже продукции, то ее нужно назвать <Продажа продукции>

Название работы должно быть по возможности кратким (не более 50 символов) и состоять из 2-3 слов. В сложных случаях также рекомендуется для каждого краткого названия работы сделать ее подробное описание, которое поместить в глоссарий.

#### **Требования к оформлению потока данных:**

1. Название потока нужно формулировать согласно следующей формуле:

Название потока = Объект, представляющий поток + Статус объекта

Примечание: Если речь идет о продукции, которую отгрузили клиенту, то поток можно назвать <Продукция, отгруженная> или <Продукция, отгруженная клиенту>. В данном случае <Продукция> это объект, представляющий поток, а <отгруженная клиенту> — статус объекта.

2. Название должно быть по возможности кратким и состоять из 2-3 слов.

## Построение DFD-модели

Построение DFD-модели базируется на принципе декомпозиции. DFD-модель включает в себя три документа, которые ссылаются друг на друга: Графические диаграммы, Миниспецификация, Словарь данных.

### 1. Контекстная диаграмма или иерархия контекстных диаграмм

Первым шагом является построение контекстной диаграммы. Диаграмма имеет звездообразную топологию, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

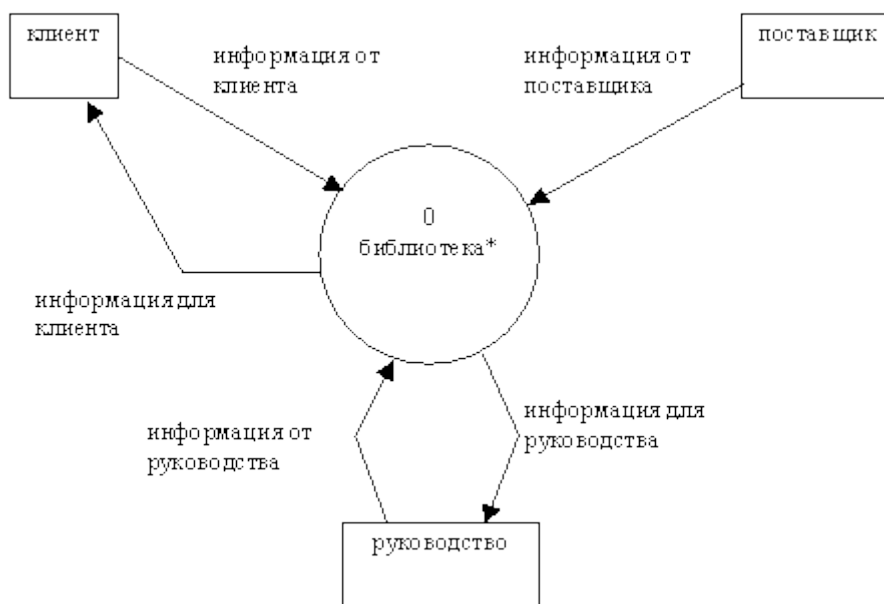


Рис. 23. Пример DFD-модели

Однако в некоторых случаях целесообразнее и нагляднее построить несколько контекстных диаграмм с иерархией:

- наличие большого количества внешних сущностей (десять и более);
- распределенная природа системы;

- многофункциональность системы с уже сложившейся или выявленной группировкой функций в отдельные подсистемы.

При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

После построения контекстных диаграмм, полученную модель следует проверить на полноту исходных данных об объектах системы и изолированность объектов (отсутствие информационных связей с другими объектами).

## 2. Детализация контекстной диаграммы

Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи диаграммы DFD. Каждый процесс, в свою очередь, может быть детализирован при помощи отдельной диаграммы или миниспецификации.

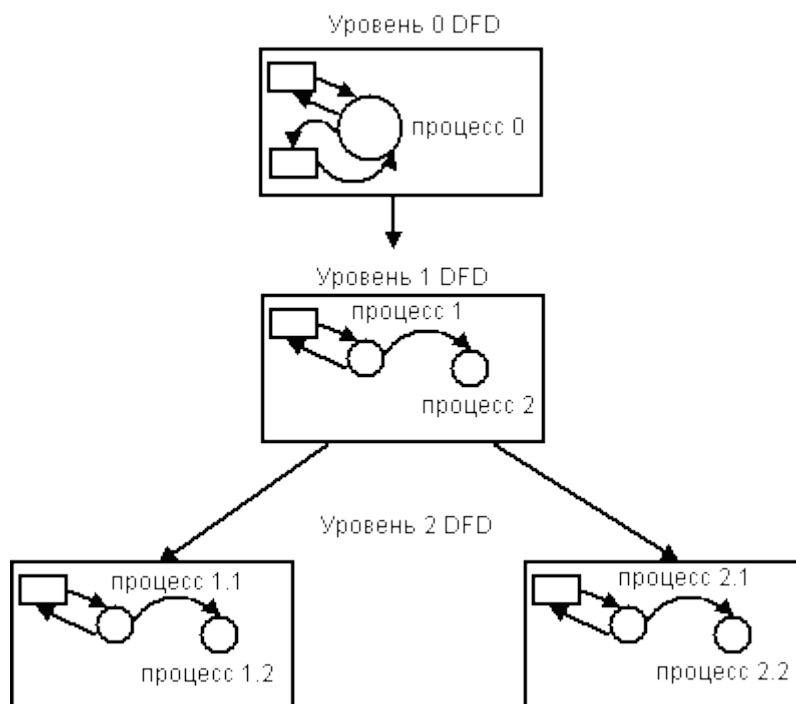


Рис. 24. Пример детализации

При детализации должны выполняться следующие правила:

- правило балансировки — при детализации процесса дочерняя диаграмма в качестве внешних источников/приемников данных может иметь только те компоненты (подсистемы, процессы, внешние

сущности, накопители данных), с которыми имеет информационную связь соответствующий процесс на родительской диаграмме;

- правило нумерации — при детализации процессов должна поддерживаться их иерархическая нумерация.
- правило семи — для того, чтобы диаграмма легко читалась, количество функций на диаграмме не должно быть больше семи.

Например, процессы, детализирующие процесс с номером 12, получают номера 12.1, 12.2, 12.3 и т.д.

### 3. Миниспецификация

Миниспецификация — документ, детально описывающий логику процесса. Она содержит номер процесса, списки входных и выходных данных, тело процесса — подробный алгоритм функции, преобразующий входные потоки данных в выходные.

Миниспецификация является конечной вершиной иерархии модели DFD. Решение о завершении детализации процесса и использовании миниспецификации принимается аналитиком исходя из следующих критериев:

- у процесса небольшое количество входных и выходных потоков данных (2-3 потока);
- процесс можно описать в виде последовательного алгоритма;
- процесс выполняет единственную логическую функцию преобразования входной информации в выходную;
- описать логику процесса можно в виде миниспецификации небольшого объема (не более 20-30 строк).

### 4. Словарь данных

В словаре данных определяется структура и содержание всех потоков данных и накопителей данных, которые присутствуют на диаграммах.

Для каждого потока в словаре хранятся: имя потока, тип, атрибуты.

Тип	Атрибут
1. Простой / групповой (объединяющий несколько потоков)	1. Имена-синонимы потока;
2. Внутренний/ внешний;	2. В случае группового потока, все потоки которые поток объединяет;
3. Поток данных/ поток управления;	3. Единицы измерения потока;

<p>4. Непрерывный (принимающий любые значения в рамках диапазона)/дискретный (принимающий конкретные значения)</p>	<p>4. Диапазон значения и типичное значение с информацией по обработке экстремальных ситуаций;</p> <p>5. Список значений и их смысл для дискретного потока;</p> <p>6. Список номеров диаграмм, в которых поток встречается;</p> <p>7. Список потоков, в которые поток входит (если в свою очередь входит в другой групповой поток);</p> <p>8. комментарии.</p>
--	--

### Проверка DFD модели

После построения законченной модели системы ее необходимо проверить на полноту и согласованность.

Модель считается полной, если все ее объекты (подсистемы, процессы, потоки данных) подробно описаны и детализированы.

Модель считается согласованной, если для всех потоков данных и накопителей данных выполняется правило сохранения информации: все поступающие куда-либо данные должны быть считаны, а все считываемые данные должны быть записаны.

### Задания к выполнению:

1. Построить диаграмму декомпозиции: «Покупатель», «Обработка заказа». Построить схему, используя двухстороннюю связь.

2. Построить диаграмму декомпозиции: «Студент», «Преподаватель», «Работать над курсовой», «Литература», «Замечания преподавателя», «Архив рефератов». Вход и выход – внешние сущности. Построить схему, используя двухстороннюю связь

### **Контрольные вопросы:**

1. Для чего служит DFD - диаграмма?
2. Основные элементы модели DFD, их описание.
3. Требования к описанию функций, потоков данных.
4. Контекстная диаграмма, иерархия контекстных диаграмм.
5. Детализация. Правила при детализации.
6. Миниспецификация. Какие критерии необходимы для принятия решения об использовании миниспецификации?
7. Словарь данных. Типы и атрибуты.
8. Когда модель считается полной и согласованной?